Embedded Software: Vision, paradigm shifts, figures, and consequences for companies in the electronics business

prof.dr.ir. Egbert-Jan Sol

The role of embedded software in most electronic products is increasing rapidly. This has far-reaching consequences for the companies involved in making these products. In this article, Egbert-Jan Sol explains what these consequences are and how to prepare for them.

Success = digital = software

How to create added value when you are in the electronics business? Computer companies, as a special branch in the electronics business, already learn how to create more added value for their customers by shifting from pure hardware to software. This transition from computer company to computerless computer company is excellently described in [1]. Since todays hardware is powerful, not scarce and therefore cheap, software is the bottleneck and good software is scarce. If your computerless computer company (i.e., your software company) can supply a better, but scarce product with less costs, you can set the price and make a profit.

What happens in the computer industry will also happen in the electronics industry. With the advance of microelectronics the electronics industry is using more and more digital hardware. Software becomes crucial for any electronics company. Old products are being replaced by newer generations which are embedded computers, repackaged with a usable user interface. Examples are PCs with word processor software which replace typewriters, the replacement of public switching telephone exchange branches based upon relay technology by large software programs running on digital computers, or tomorrow's High Definition TeleVision (HDTV) which is based on the same technology as a (UNIX) workstation.

This transition towards a 'digital world' will be one of the triggers for the fifth Kondratieff (long-term) economic wave. Kondratieff's waves are the long (40+ years) economical waves. Presently we are at the bottom between the fourth and the fifth. They are associated with industrial revolutions as during a new wave new technology finally enables mankind to do (new) things in a different and better way. These *things* might be well-known activities, such as listening to music, or new ones, such as e-mail/fax with ubiquitous computing [2].

Successful companies will be those who find ways to create (new) value for their users. They are better served/equipped to do things with less costs like the computerless computer company described above.

In the next decade three core competences are crucial for the electronics business to remain in a competitive position: low cost hardware manufacturing, user interfacing, and (embedded) software. To be successful in low cost hardware, one has to be successful in management, such as lean manufacturing. User interfacing cannot be successfully managed or even engineered today. To build up a core competence in this area, artists and craftspersons should be attracted in this phase. Software is a young discipline. In its present state, success is determined by excellent engineering. Those who will be successful in the future are the ones who bring (embedded) software on the level that it can be managed as soon as possible. The question is how.

Software: unmanageable?

Today, software is said to be unmanageable. Software is late, it costs too much, etc. Many of these managerial problems are related to the fact that software does not reach the required productivity improvements. With insufficient growth in productivity software engineers act as fire fighters. They do not get the opportunity to improve their work in a structured manner. This lack of sufficient productivity improvement results in a larger share for software in the cost structure.

The price/performance of hardware doubles ev-

ery 1.5 year (i.e., an increase with a factor 100 in ten years), while software productivity doubles, on average, only every five years (i.e., a price/performance ratio increase with a factor 4 in ten years). At the same time it is not uncommon that software demand, due to new hardware features, increases by a factor 4 every five years (16 × more costs in ten years). The hardware costs thus drop with a factor 100 in ten years, while software costs grow with a factor 4 in the same period.

Imagine a certain product for which hardware costs today are 50% and its software costs 10%. The remaining 40% is not changing. Within a decade software becomes 40% of the (original) total price, while hardware and remaining costs become 40.5%. In total the product price drops 20%. Underneath this apparently not so disturbing 20% cost reduction in 10 years a complete paradigm shift or mind set transition occurred. The software share rises from 10% to 40%. As a result the software manager will continuously lack sufficient resources, his software is always late and software is said to be unmanageable.

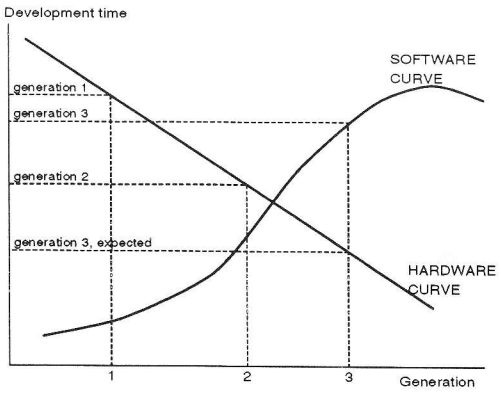


Figure 1: Illustration of software becoming a bottleneck in time-to-market

Figure 1 illustrates that the result of this will be that hardware is flexible, and that software becomes a bottleneck. Everyone expects the improvements in time-to-market to follow previous successes in hardware, but the figure illustrates how expected results are realistic only for the first two generations which contain hardly any software and where hardware development lead times are critical. For the third generation software suddenly becomes the bottleneck and the development time grows instead of reduces. Software becomes rigid in the sense that hardware can be changed overnight while software changes can take months to be processed.

Getting software under control

How to get software under control such that productivity improves as needed? One initiative is to get the right people and build-up a core competence. This approach takes years, requires the right selection skills and some patience.

Another initiative can be the use of better tools. In the software engineering world, this implies faster computer systems and better CASE tools. In embedded software in particular lower level CASE tools are needed. Examples are advanced emulators, high-speed down-loading, automatic testing, simple configuration and version control, etc.

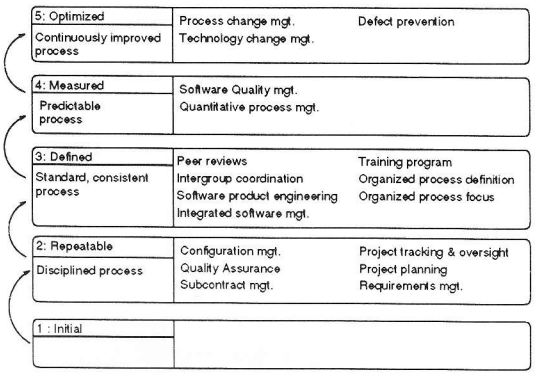


Figure 2: Sketch of Humphrey's Capability Maturity Model (CMM)

A third initiative is based upon improving the organization of the software development process. In this case, process models are important. A well-known example in the embedded software world is the capability maturity model (also known as Humphrey's CMM [3], see Figure 2). It is used as an assessment method to determine the capability maturity of a software development organization. It does not describe a software development process, nor does it guarantee that a higher productivity will be reached at higher maturity levels. Its main benefit is that it forces organizations to focus on their processes and to improve them. The assumption is that a better process focus will improve the quality of the delivered end product and make the software development process more manageable. Once more manageable, it is expected that higher productivity levels can be reached by continuously monitoring the process, measuring it, and improving it.

As the Humphrey model is well-suited for embedded software only, the electronics industry often applies concurrent engineering processes these days. By simultaneously developing the mechanics (e.g., plastic housing), electronic hardware (e.g., ASICs and boards), and the embedded software the time-to-market can be reduced. Research is working on newer models such as the EUT-RACE model [4], shown in Figure 3. This model is better suited to improve the overall product creation process since it does not only focus on specific software (engineering) aspects but also on more general management aspects like leadership, team formation, discipline, etc.

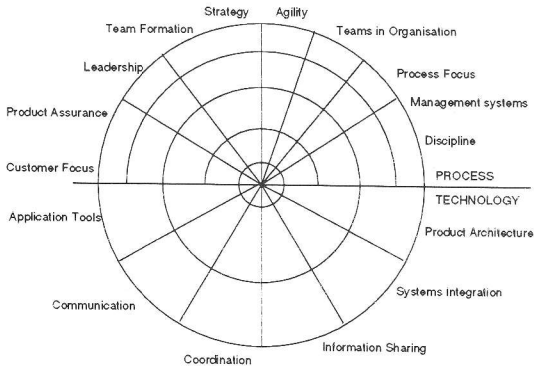


Figure 3: The EUT-RACE model

The fourth initiative is improving productivity by doing things smarter. The objective of getting software development under control and the strategy to improve productivity implies that an organization should excel in its learning behavior. Understanding the present mechanism and productivity gap as well as simultaneously applying all four initiatives (bright people, good tools, process focus, and learning organizations) will contribute in achieving a more manageable software environment.

If you are not willing or able to build up a core competence on your own because it takes too long or costs too much there always is a fifth initiative, which is getting someone else involved. In this time of leaner organizations one has to understand what the core competences of the organization are [5, 6, 7] and what should be outsourced to a co-maker. Embedded software can be one of the outsourced activities. In the resulting partnership it must be clear that the co-maker has a proven capability in the sense that he can develop software at a high level of quality (e.g., ISO 9000 certified in software development or on level 2 or higher of the CMM) and at a high productivity rate.

The old Chinese saying "you are living in an interesting time" implies that life is not easy. Embedded software developers are living in an interesting time. Being structurally overtime and over budget, they are mostly fire-fighting, but they learn rapidly and embedded software is no longer a back-yard activity in a development lab. It has become a main activity and is becoming better understood. Within several years embedded software engineering will be professionally manageable. That is, you can get it under control, whether developed in-house or in a co-maker relationship. □

References

[1] Rappaport & Halevi, Computerless Computer Company, Harvard Business Review, July 1991

[2] Mark Weiser, The Computer for the 21st Century, Scientific American, September 1991

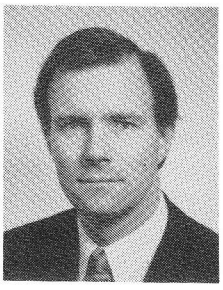
[3] Humphrey, W.S., Managing the Software Process, 1989, Addison-Wesley

[4] De Graaf & Sol, Assessing Europe's Readiness for Concurrent Engineering, Proceedings 1st Int. Conf. on Concurrent Engineering, Research & Applications, Pittsburgh, 1994

[5] Prahalad & Hamel, The core competence of the corporation, Harvard Business Review, May 1990

[6] Stalk et al., Competing on capabilities, Harvard Business Review, March 1992

[7] Normann & Ramirez, From value chain to value constellation, Harvard Business Review, July 1993



Prof.dr.ir. Egbert-Jan Sol is a consultant at BSO/Origin Eindhoven. He is also part-time professor at the department of Industrial Engineering & Management Sciences of Eindhoven University of Technology.